

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A networked computer comprising an unsolicited message diverting communications processor connected between mail transfer agents MTA\_0 with an Internet address IP\_0, a from-address A\_0, a declared domain D\_0, and a real domain DD\_0, and MTA\_1 with an Internet address IP\_1, a domain D\_1, a to-address A\_1, a diversion address A'\_1, and a save\_spam database, the unsolicited message diverting communications processor comprising:

- a) monitoring means for monitoring the communications between MTA\_0 and MTA\_1;
- b) determining means for determining if the communications contains an unsolicited message; and
- c) intercepting means for intercepting a RCPT reply from MTA\_0, substituting the diversion address A'\_1 for the to-address A\_1 in the RCPT reply and sending a modified RCPT reply to MTA\_1 if the message is determined to be unsolicited and if the to-address A\_1 is in the save\_spam database;

wherein the unsolicited message diverting communications processor does not intercept communications between MTA\_0 and MTA\_1 before a RCPT command from MTA\_0 is received by the unsolicited message diverting communications processor, and

wherein the connection with MTA\_0 is rejected before the data portion of the unsolicited message is transmitted.

2. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes an allow\_address database and wherein the determining means determines if a message is not unsolicited by checking if the IP\_0 is in the allow\_address database.

3. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes a prevent\_address database and wherein

the determining means determines if a message is unsolicited by checking if IP\_0 is in the prevent\_address database.

4. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes access to an open relay database and wherein the determining means determines if a message is unsolicited by checking if IP\_0 is in the open relay database.

5. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes access to a DNS (domain name server) database and wherein the determining means determines if a message is unsolicited by checking if IP\_0 has a domain name entry DD\_0 in the DNS database.

6. ((Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes a bad\_from database and wherein the determining means determines if a message is unsolicited by checking if the from-address A\_0 is in the bad\_from database.

7. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes a suspect\_domain database and wherein the determining means determines if a message is unsolicited by checking if the real domain DD\_0 matches the domain of the from-address A\_0 and the domain of the from-address A\_0 is in the suspect\_domain database.

8. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the from-address A\_0 matches the to-address A\_1.

9. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the declared domain D\_0 of MTA\_0 is the same as the domain D\_1 of MTA\_1.

10. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the declared domain D\_0 of MTA\_0 does not match the real domain DD\_1DD\_0 and the declared domain D\_0 is in the suspect\_domain database.

11. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes a no\_filter database and wherein the determining means determines if an unsolicited message should be blocked by checking if the to-address A\_1 is in the no\_filter database.

12. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes a rejected\_connection database which logs the time, from-address A\_0, to-address A\_1, and the reason for the rejection if a message is rejected if the message is determined to be unsolicited.

13. (Currently amended) The networked computer ~~unsolicited message diverting communications processor~~ in Claim 1, further includes an allowed\_connection database which logs the time and to-address A\_1 if the message is determined not to be unsolicited.

14. (Currently amended) A method implemented in a networked computer for a receiving networked computer system with an Internet connection, a mail transfer agent MTA\_1, an Internet address IP\_1, a to-address A\_1, a diversion address A'\_1, a save\_spam database and an operating system capable of executing the method to divert unsolicited messages from a transmitting networked computer system with an Internet connection and a mail transfer agent MTA\_0, an Internet address IP\_0, a from-address A\_0, a declared domain D\_0, and a real domain DD\_0 comprising the steps of:

- a) waiting for a new SMTP connection request;
- b) relaying and monitoring the replies from MTA\_0 to MTA\_1;
- c) relaying replies from MTA\_1 to MTA\_0;
- d) intercepting a RCPT reply from MTA\_0 to MTA\_1;
- e) determining if the message is unsolicited by analyzing the monitored replies;
- f) releasing the intercepted RCPT reply if the message is determined not to be

unsolicited; and

g) substituting the diversion address A'\_1 for the to-address A\_1 in the RCPT reply and sending the modified reply to MTA\_1 if the message is determined to be unsolicited and if the to-address A\_1 is in the save\_spam database;

whereby MTA\_1 controls the interaction between MTA\_0 and MTA\_1 before a RCPT command from MTA\_0 is received and whereby the connection with MTA\_0 is rejected before the data portion of the unsolicited message is transmitted.

15. (Currently amended) A method implemented in a networked computer system for a receiving networked computer system with an Internet connection, a mail transfer agent MTA\_1, an IP address IP\_1, a domain name D\_1, a to-address, A\_1, a recipient diversion address A'\_1, an allow\_address database, a prevent\_address database, a suspect\_domain database, a bad\_from database, a no\_filter database, a rejected\_connection database, an allowed\_connection database, a save\_spam database, a diversion database, and an operating system capable of executing the method to divert unsolicited messages from a transmitting networked computer system with an Internet connection, a mail transfer agent MTA\_0, an IP address of IP\_0, a declared domain name D\_0, a real domain name DD\_0, and a from-address of A\_0 comprising the steps of:

a) waiting for a SMTP connection request on the receiving networked computer system's Internet connection;

b) sending a 220 reply to MTA\_0 to acknowledge the requested SMTP connection request;

c) extracting IP\_0 from the SMTP connection request;

d) requesting the real domain name DD\_0 for IP\_0 from a DNS database;

e) testing if the real domain name DD\_0 is "no name";

f) testing if IP\_0 is in an open relay database;

g) testing if IP\_0 is in the allow\_address database;

h) testing if IP\_0 is in the prevent\_address database,

i) requesting a connection with MTA\_1;

j) waiting for a 220 reply from MTA\_1 to acknowledge the requested connection;

k) waiting for a reply from either MTA\_0 or MTA\_1;

l) jumping to step o) if the reply is not from MTA\_1;

- m) relaying the reply from MTA\_1 to MTA\_0;
- n) jumping to step k) to wait for a new reply;
- o) jumping to step u) if the reply from MTA\_0 is not a HELO;
- p) extracting the declared domain name D\_0 from the reply;
- q) testing if the declared domain name D\_0 is the same as D\_1;
- r) testing if the declared domain name D\_0 of MTA\_0 does not match the real domain name DD\_0 of MTA\_0 AND the declared domain name D\_0 of MTA\_0 is in the suspect\_domain database;
- s) relaying the HELO reply from MTA\_0 to MTA\_1;
- t) jumping to step k) to wait for a new reply;
- u) jumping to step aa) if reply from MTA\_0 is not a MAIL;
- v) extracting the from-address A\_0;
- w) testing if A\_0 is in the bad\_from database;
- x) testing if DD\_0 does not match the domain of A\_0 and the domain of A\_0 is in the suspect\_domain database;
- y) relaying MAIL reply to MTA\_1;
- z) jumping to step k) to wait for a new reply;
- aa) jumping to step qq) if the reply from MTA\_0 is not a RCPT;
- bb) extracting the to-address A\_1;
- cc) testing if A\_1 is in the no\_filter database;
- dd) testing if A\_0 matches A\_1;
- ee) jumping to step nn) if t\_allow OR t\_no\_filter OR NOT ( t\_prevent OR t\_open OR t\_DD\_0 OR t\_bad\_from OR t\_suspect\_domain OR t\_match );
- ff) logging time, A\_0, A\_1, and reason for rejection in the rejected\_connection database;
- gg) jumping to step ll) if A\_1 is not in the save\_spam database;
- hh) looking up A'\_1 in the diversion database;
- ii) substituting A'\_1 for A\_1 in the RCPT reply;
- jj) sending the modified RCPT reply to MTA\_1;
- kk) jumping to step k) to wait for a new reply;
- ll) rejecting the connection with MTA\_0 connection by sending a 550 reply to MTA\_0;
- mm) jumping to step k) to wait for a new reply;

nn) logging time and A\_1 in the allowed\_connection database;  
oo) relaying RCPT reply from MTA\_0 to MTA\_1;  
pp) jumping to step k) to wait for a new reply;  
qq) jumping to step bbb) if the reply from MTA\_0 is not DATA;  
rr) relaying DATA reply to MTA\_1;  
ss) waiting for a 354 reply from MTA\_1;  
tt) relaying the 354 reply from MTA\_1 to MTA\_0;  
uu) waiting for the data from MTA\_0;  
vv) relaying the data from MTA\_0 to MTA\_1;  
ww) waiting for a .r\n from MTA\_0;  
xx) relaying the .r\n from MTA\_0 to MTA\_1;  
yy) waiting for a 250 reply from MTA\_1;  
zz) relaying the 250 reply to MTA\_0;  
aaa) jumping to step k) to wait for a new reply;  
bbb) jumping to step eee) if reply from MTA\_0 is not RSET, SEND, SCML, SAML, VRFY, NOOP, EXPN, HELP, or TURN;  
ccc) relaying the reply to MTA\_1;  
ddd) jumping to step e) to wait for a new reply;  
eee) jumping to step jjj) if the reply from MTA\_0 is not a QUIT;  
fff) relaying the QUIT reply to MTA\_1;  
ggg) waiting for a 221 reply from MTA\_1;  
hhh) relaying the 221 reply from MTA\_1 to MTA\_0;  
iii) jumping to step a) to wait for a new connection;  
jjj) sending a 500 reply to MTA\_0 to signal a syntax error; and  
kkk) jumping to step a) to wait for a new connection.

16. (Cancelled)

17. (Cancelled)